



Microsoft®  
**SQL Server® 2008**

## **Section 5**

Your Data, Any Place,  
Any Time

# Mathematical functions

- Select **ABS(-11.5)**                      --returns 11.5
- Select **CEILING(11.3)**                      --returns 12
- Select **FLOOR(11.6)**                      --returns 11
- Select **POWER(2,4)**                      --returns 16
- Select **SQUARE(9)**                      --returns 81
- Select **SQRT(81)**                      --returns 9
- Select **RAND()**                      --returns random value from 0 to 1.
- Select **(RAND()\*100)**                      ???!
- Select **FLOOR(RAND()\*100)**                      ???!

# Mathematical functions

- What is the result ?

```
Declare @c int
Set @c=1
While (@c<=10)
Begin
    Print floor ( Rand() *100 )
    Set @c=@c+1
End
```

# Mathematical functions

## ■ ROUND() Function

- `Select ROUND( 850.556 , 2 )` --returns 850.560
- `Select ROUND( 850.556 , 2 , 1 )` --returns 850.550
- `Select ROUND( 850.556 , 1 )` --returns 850.600
- `Select ROUND( 850.556 , 1 , 1 )` --returns 850.500
- `Select ROUND( 850.556 , -2 )` --returns 900.000
- `Select ROUND( 850.556 , -1 )` --returns 850.000

# User defined functions(UDF)

- In SQL server there are 3 types of User Defined Functions
  - Scalar functions.
  - Inline table-valued functions.
  - Multi-statement table-valued functions.

# 1. Scalar UDF

- may or may not have parameter but always return single value.
- Returns any type except text, ntext, Image, Cursor and timestamp.

```
CREATE FUNCTION Fun_name (@parm1 datatype, @param2 datatype)
RETURNS returned_type
AS
BEGIN
    --function Body
    Return return ed_Value
END
```

# 1. Scalar UDF

```
CREATE FUNCTION fn_CalculateAge (@DOB date)
RETURNS int
AS
BEGIN
    Declare @age int
    SET @age=DateDiff( YEAR , @DOB , GETDATE() ) -
        CASE
            WHEN (MONTH (@DOB) > MONTH (GETDATE ())) OR
                (MONTH (@DOB) =MONTH (GETDATE ())) AND
                (DAY(@DOB) > DAY(GETDATE ()))
            THEN 1
            ELSE 0
        END
    Return @age
END
```

```
Select sname ,sDOB, dbo. CalculateAge (sDOB) from tbl_students
```

# Assignment

Can you transfer the previous  
function into procedure  
And use it Like function  
demonstrate your answer??



# 2. Inline table valued functions

- Returns a table.

```
CREATE FUNCTION fn_studentByDeptId (@dept int)  
RETURNS TABLE  
AS  
RETURN ( SELECT sname, sDOB , sdept  
From tbl_Students  
Where sdept=@dept)
```

```
SELECT * From fn_studentByDeptId (1)
```

```
SELECT * From fn_studentByDeptId (1)  
Where sgender='male'
```

# Assignment

- Explain and show what is the result ???

```
SELECT s_name, s_DOB , d_name  
From fn_studentByDeptId (1) S  
JOIN tbl_depts D  
On D.d_id=s.s_dept  
Where s_gender='male'
```



Eng.Hamada Rizk